上級編

第42講 素因数分解

例題

以下の条件で，整数 number の約数を効率的に算出して列挙するプログラムを作成する。

・約数とは，ある整数を割ることができ商が整数になる数のことである。

・Yakusuu.sort() という命令は，配列 Yakusuu の要素を昇順に並べ替えるものである。

以下のプログラムの空欄に入る最も適切なものを，あとの①～④のうちから一つ選びなさい。

```
number = 36
Yakusuu = [] # 求めた約数を格納する配列

for i in range(1, int(number ** 0.5) + 1): # number ** 0.5 は，number の平方根
  if ____________ :
    Yakusuu.append(i) # 約数 i を Yakusuu 配列に追加する
    pair_yakusuu = number // i # 約数 i のペア約数
    if i != pair_yakusuu:
      Yakusuu.append(pair_yakusuu)

Yakusuu.sort() # 配列内の要素を昇順に並べ替える
print(number, 'の約数：', Yakusuu)
```

① `number % i != 0`

② `number % i == 0`

③ `number // i == 1`

④ `number % i == 1`

上級編　第42講　確認問題

入力された整数を素因数分解し，結果を効率的に列挙するプログラムを作成する。

なお，素因数分解とは，与えられた整数をその数を構成する素数の積に分解することである。

以下のプログラムの空欄に入る最も適切なものを，あとの①～④のうちから一つ選びなさい。

```
number = int(input('素因数分解を行う整数を入力してください。'))

original_number = number # numberの初期値を保存しておく
i = 2 # 素数の候補を一時的に格納（2は最小の素数）
Factors = [] # 因数分解した値を格納する配列

while i * i <= number:
  if [            ]:
    i = i + 1
  else:
    number = number // i
    Factors.append(i) # 引数の値を配列Factorsに追加する
if number > 1: # 空の配列にならないための条件
  Factors.append(number)

print(original_number, 'の素因数分解：', Factors)
```

① `number % i != 0`

② `number % i == 0`

③ `number / i > 0`

④ `number // i == 1`

上級編　例題　解答

第34講　コインの枚数

例題　解答：④

第35講　駐車料金の判定

例題　解答：④

第36講　割引のある料金

例題　解答：④

第37講　テストの集計

例題　解答：①

第38講　じゃんけん

例題　解答：①

第39講　サイコロの出目

例題　解答：③

第40講　中央値の判定

例題　解答：①

第41講　素数判定

例題　解答：③

第42講　素因数分解

例題　解答：②

第43講　倍数の判定・FizzBuzz

例題　解答：④

第44講　バブルソート・選択ソート

例題　解答：③

第45講　グラフ理論

例題　解答：②

第46講　待ち行列

例題　解答：②

第47講　パリティチェック

例題　解答：①

第48講　すごろくゲーム

例題　解答：④

上級編　第 42 講　確認問題　解答

正解：①

解説：

8 行目からの if 文では，if 条件式が False であるとき（10 行目の else 文の処理にうつるとき），素数を表す変数 i を配列 Factors の末尾に追加している。ここから，else 文では，変数 i が入力した値の素因数である場合を条件としていることが分かる。

つまり，8 行目の if 文では，変数 i がその整数の素因数でないときの条件式を指定すればよい。よって，正解は ① number % i != 0 である。

```
01 number = int(input('素因数分解を行う整数を入力してください。'))
02 
03 original_number = number # number の初期値を保存しておく
04 i = 2 # 素数の候補を一時的に格納（2 は最小の素数）
05 Factors = [] # 因数分解した値を格納する配列
06 
07 while i * i <= number:
08   if number % i != 0:
09     i = i + 1
10   else:
11     number = number // i
12     Factors.append(i) # 引数の値を配列 Factors に追加する
13 if number > 1: # 空の配列にならないための条件
14   Factors.append(number)
15 
16 print(original_number, 'の素因数分解：', Factors)
```

実行結果

```
素因数分解を行う整数を入力してください。10
10 の素因数分解： [2, 5]
```